

Laborator nr. 6 Tablouri bidimensionale - matrici

A. Probleme rezolvate:

Problema 1: Sa se scrie un program care calculeaza produsul a doua matrici (tablouri bidimensionale) $A_{n \times m}$, $B_{m \times p}$.

Solutie:

```
#include <iostream.h>
int n,m,p,i,j,k,a[10][10],b[10][10],c[10][10];
int main(void)
{
    cout<<"Dati dimensiunile matricei A"<<endl;
    cin>>n;
    cin>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++) cin>>a[i][j];
    cout<<"Elementele matricei A sunt: "<<endl;
    for(i=1;i<=n;i++){
        for(j=1;j<=m;j++) cout<<a[i][j]<<" ";
        cout<<endl;
    }
    cout<<"Dati dimensiunile matricei B"<<endl;
    cin>>m;
    cin>>p;
    for(i=1;i<=m;i++)
        for(j=1;j<=p;j++) cin>>b[i][j];
    cout<<"Elementele matricei B sunt: "<<endl;
    for(i=1;i<=m;i++){
        for(j=1;j<=p;j++) cout<<b[i][j]<<" ";
        cout<<endl;
    }
    for(i=1;i<=n;i++)
        for(j=1;j<=p;j++)
            for(k=1;k<=m;k++)
                c[i][j]+=a[i][k]*b[k][j];
    cout<<"Elementele matricei produs "<<endl;
    for(i=1;i<=n;i++){
        for(j=1;j<=p;j++) cout<<c[i][j]<<" ";
        cout<<endl;
    }
}
```

Laborator - Programarea Calculatoarelor si Limbaje de Programare (2022) Limbajul C++

Adrian Runceanu

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

Online C++ Compile

```
1 #include <iostream>
2 using namespace std;
3 int n,m,p,l,j,k,a[10][10],b[10][10],c[10][10];
4 int main(void)
5 {
6     cout<<"Dati dimensiunile matricii A\n";
7     cin>>n;
8     cin>>m;
9     for(i = 1; i <= n; i++)
10         for(j = 1; j <= m; j++)
11             cin>>a[i][j];
12     cout<<"Elementele matricii A sunt: \n";
13     for(i = 1; i <= n; i++)
14     {
15         for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
16         cout<<"\n";
17     }
18     cout<<"Dati dimensiunile matricii B\n";
19     cin>>m;
20     cin>>p;
21     for(i = 1; i <= m; i++)
22         for(j = 1; j <= p; j++)
23             cin>>b[i][j];
24     cout<<"Elementele matricii B sunt: \n";
25     for(i = 1; i <= m; i++)
26     {
27         for(j = 1; j <= p; j++) cout<<b[i][j]<<" ";
28         cout<<"\n";
29     }
30
31     for(i = 1; i <= n; i++)
32         for(j = 1; j <= p; j++)
33             for(k = 1; k <= m; k++)
34                 c[i][j] += a[i][k] * b[k][j];
35     cout<<"Elementele matricii produs: \n";
36     for(i = 1; i <= n; i++)
37     {
38         for(j = 1; j <= p; j++) cout<<c[i][j]<<" ";
39         cout<<"\n";
40     }
41     return 0;
42 }
```

Execute Mode: Version, Inputs & Arguments

GCC 9.1.0

☐ Interactive

Stdin Inputs

2 3

1 2 3

4 5 6

3 4

1 2 3 4

5 6 7 8

9 10 11 12

CommandLine Arguments

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3316 kilobytes)

Dati dimensiunile matricii A

Elementele matricii A sunt:

1 2 3

4 5 6

Dati dimensiunile matricii B

Elementele matricii B sunt:

1 2 3 4

5 6 7 8

9 10 11 12

Elementele matricii produs

38 44 50 56

83 98 113 128

Problema 2: Sa se scrie un program care calculeaza minimul și maximul dintr-o matrice cu n linii și m coloane ($A_{n \times m}$) ($1 \leq n, m \leq 30$).

Solutie:

```
#include <iostream>
using namespace std;
int n, m, minim, i, j, maxim, a[30][30];
int main(void)
{
    cout<<"Dati dimensiunile matricei A\n";
    cin>>n;
    cin>>m;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++) cin>>a[i][j];
    cout<<"Elementele matricei A sunt: \n";
    for(i = 1; i <= n; i++){
        for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
        cout<<'\\n';
    }
    minim = a[1][1];
    maxim = a[1][1];
    for(j = 1; j <= m; j++){
        if(a[1][j] < minim) minim = a[1][j];
        if(a[1][j] > maxim) maxim = a[1][j];
    }
    for(i = 2; i <= n; i++)
        for(j = 1; j <= m; j++){
            if(a[i][j] < minim)    minim = a[i][j];
            if(a[i][j] > maxim)    maxim = a[i][j];
        }
    cout<<"\\nElementul minim din matrice este "<<minim;
    cout<<"\\nElementul maxim din matrice este "<<maxim;
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 int n, m, minm, i, j, maxm, a[30][30];
4 int main(void)
5 {
6     cout<<"Dati dimensiunile matricei A\n";
7     cin>>n;
8     cin>>m;
9     for(i = 1; i <= n; i++)
10         for(j = 1; j <= m; j++)
11             cin>>a[i][j];
12     cout<<"Elementele matricei A sunt: \n";
13     for(i = 1; i <= n; i++)
14     {
15         for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
16         cout<<"\n";
17     }
18     minm = a[1][1];
19     maxm = a[1][1];
20     for(j = 1; j <= m; j++)
21     {
22         if(a[1][j] < minm) minm = a[1][j];
23         if(a[1][j] > maxm) maxm = a[1][j];
24     }
25     for(i = 2; i <= n; i++)
26         for(j = 1; j <= m; j++)
27         {
28             if(a[i][j] < minm) minm = a[i][j];
29             if(a[i][j] > maxm) maxm = a[i][j];
30         }
31     cout<<"\nElementul minm din matrice este " <<minm;
32     cout<<"\nElementul maxm din matrice este " <<maxm;
33     return 0;
34 }
```

Execute Mode, Variables, Inputs & Arguments

Compiler: GCC 9.1.0 Interactive Stdin Inputs

CommandLine Arguments

2 3
10 2 -3
4 15 6

Execute

Result
CPU Time: 0.00 sec(s), Memory: 1388 kilobyte(s)

```
Dati dimensiunile matricei A
Elementele matricei A sunt:
10 2 -3
4 15 6

Elementul minm din matrice este -3
Elementul maxm din matrice este 15
```

Problema 3: Sa se scrie un program care calculeaza transpusa unei matrici $A_{n \times m}$ ($1 \leq n, m \leq 30$).

Solutie:

```
#include <iostream>
using namespace std;
int n, m, i, j, a[30][30], b[30][30];
int main(void)
{
    cout<<"Dati dimensiunile matricei A\n";
    cin>>n;
    cin>>m;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++) cin>>a[i][j];
    cout<<"Elementele matricei A sunt: \n";
    for(i = 1; i <= n; i++){
        for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
    }
```

```
        cout<<"\n";
    }
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++) b[j][i] = a[i][j];
    cout<<"Matricea transpusa este \n";
    for(i = 1; i <= m; i++){
        for(j = 1; j <= n; j++)    cout<<b[i][j]<<" ";
        cout<<"\n";
    }
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

Online C++ Compiler IDE

```
1 #include <iostream>
2 using namespace std;
3 int m, n, i, j, a[30][30], b[30][30];
4 int main(void)
5 {
6     cout<<"Dati dimensiunile matricei A\n";
7     cin>>m;
8     cin>>n;
9     for(i = 1; i <= n; i++)
10         for(j = 1; j <= m; j++)
11             cin>>a[i][j];
12     cout<<"Elementele matricei A sunt: \n";
13     for(i = 1; i <= n; i++)
14     {
15         for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
16         cout<<"\n";
17     }
18     for(i = 1; i <= n; i++)
19         for(j = 1; j <= m; j++)
20             b[j][i] = a[i][j];
21     cout<<"Matricea transpusa este \n";
22     for(i = 1; i <= m; i++)
23     {
24         for(j = 1; j <= n; j++)    cout<<b[i][j]<<" ";
25         cout<<"\n";
26     }
27     return 0;
28 }
29
```

Execute Mode, Version, Inputs & Arguments

GCC 9.1.0 ☐ Interactive Stdin Inputs

CommandLine Arguments

2 3
1 2 3
4 5 6

Result

CPU Time: 0.00 sec(s), Memory: 3256 kb(byte(s))

```
Dati dimensiunile matricei A
Elementele matricei A sunt:
1 2 3
4 5 6
Matricea transpusa este
1 4
2 5
3 6
```

Problema 4: Se considera o matrice $A_{n \times n}$. Sa se calculeze:

- suma elementelor de pe diagonala principala
- produsul elementelor de pe diagonala secundara
- minimele din elementele aflate deasupra, respectiv sub diagonala principala
- maximele din elementele aflate deasupra, respectiv sub diagonala secundara

Solutie:

```
#include <iostream>
using namespace std;
int n, i, j, a[30][30], min1, min2, max1, max2, suma, produs;
int main(void)
{
    cout<<"Dati dimensiunile matricei A\n";
    cin>>n;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++) cin>>a[i][j];
    cout<<"Elementele matricei A sunt: \n";
    for(i = 1; i <= n; i++){
        for(j = 1; j <= n; j++) cout<<a[i][j]<<" ";
        cout<<'\\n';
    }
    // ===== suma elementelor de pe diagonala principala =====
    suma = 0;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++)
            if(i == j) suma += a[i][j];
    cout<<"\\nSuma elementelor de pe diagonala principala este
"<<suma;

    // ===== produsul elementelor de pe diagonala secundara =====
    produs = 1;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++)
            if(i + j == n + 1) produs *= a[i][j];
    cout<<"\\nProdusul elementelor de pe diagonala secundara
este "<<produs;

    // ===== minimul elementelor de deasupra diagonalei principale =
    min1 = 32768;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++)
            if(i < j)
```

```
        if(min1 > a[i][j]) min1 = a[i][j];
    cout<<"\nMinimul de deasupra diag. principale este
"<<min1;

    // ===== minimul elementelor de sub diagonala principala =====
    min2 = 32768;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++)
            if(i > j)
                if(min2 > a[i][j]) min2 = a[i][j];
    cout<<"\nMinimul de sub diag. principala este "<<min2;

    // == maximul elementelor de deasupra diagonalei secundare ==
    max1 = -32768;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++)
            if(i + j < n+1)
                if(max1 < a[i][j]) max1 = a[i][j];
    cout<<"\nMaximul de deasupra diag. secundare este
"<<max1;

    // ===== maximul elementelor de sub diagonala secundare =====
    max2 = -32768;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= n; j++)
            if(i + j > n+1)
                if(max2 < a[i][j]) max2 = a[i][j];
    cout<<"\nMaximul de sub diag. secundare este "<<max2;
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1  #include <iostream>
2  using namespace std;
3  int n, i, j, a[30][30], min1, min2, max1, max2, suma, produs;
4
5  int main(void)
6  {
7      cout<<"Dati dimensiunile matricei A\n";
8      cin>>n;
9      for(i = 1; i <= n; i++)
10         for(j = 1; j <= n; j++)
11             cin>>a[i][j];
12      cout<<"Elementele matricei A sunt: \n";
13      for(i = 1; i <= n; i++)
14      {
15         for(j = 1; j <= n; j++) cout<<a[i][j]<<" ";
16         cout<<"\n";
17     }
18     // ===== suma elementelor de pe diagonala principala =====
19     suma = 0;
20     for(i = 1; i <= n; i++)
21         for(j = 1; j <= n; j++)
22             if(i == j) suma += a[i][j];
23     cout<<"\nSuma elementelor de pe diagonala principala este "<<suma;
24
25     // ===== produsul elementelor de pe diagonala secundara =====
26     produs = 1;
27     for(i = 1; i <= n; i++)
28         for(j = 1; j <= n; j++)
29             if(i + j == n + 1) produs *= a[i][j];
30     cout<<"\nProdusul elementelor de pe diagonala secundara este "<<produs;
31
32     // ===== minimul elementelor de deasupra diagonalei principale =
33     min1 = 32768;
34     for(i = 1; i <= n; i++)
35         for(j = 1; j <= n; j++)
36             if(i < j)
37                 if(min1 > a[i][j]) min1 = a[i][j];
38     cout<<"\nMinimul de deasupra diag. principale este "<<min1;
39
40     // ===== minimul elementelor de sub diagonala principale =====
41     min2 = 32768;
42     for(i = 1; i <= n; i++)
43         for(j = 1; j <= n; j++)
44             if(i > j)
45                 if(min2 > a[i][j]) min2 = a[i][j];
46     cout<<"\nMinimul de sub diag. principale este "<<min2;
47
48     // ===== maximul elementelor de deasupra diagonalei principale =====
49     max1 = -32768;
50     for(i = 1; i <= n; i++)
51         for(j = 1; j <= n; j++)
52             if(i < j)
53                 if(max1 < a[i][j]) max1 = a[i][j];
54     cout<<"\nMaximul de deasupra diag. principale este "<<max1;
55
56     // ===== maximul elementelor de sub diagonala principale =====
57     max2 = -32768;
58     for(i = 1; i <= n; i++)
59         for(j = 1; j <= n; j++)
60             if(i > j)
61                 if(max2 < a[i][j]) max2 = a[i][j];
62     cout<<"\nMaximul de sub diag. principale este "<<max2;
63     return 0;
64 }
65
```




```
Dati dimensiunile matricei A
Elementele matricei A sunt:
1 2 3
4 5 6
7 8 9
Suma elementelor de pe diagonala principala este 15
Produsul elementelor de pe diagonala secundara este 108
Minimal de deasupra diag. principale este 2
Minimal de sub diag. principala este 4
Maximal de deasupra diag. principale este 6
Maximal de sub diag. principala este 8
```

Problema 5: Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) cu elemente numere intregi. Sa se determine linia (liniile) din matrice care contine cele mai multe elemente nenule.

Solutie:

```
#include <iostream>
using namespace std;
int a[30][30], n, m, i, j, maxim, numar;
int main(void)
{
    cout<<"Dati dimensiunile matricei A\n";
    cin>>n>>m;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++) cin>>a[i][j];
    cout<<"Elementele matricei A sunt: \n";
    for(i = 1; i <= n; i++){
        for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
        cout<<"\n";
    }
    maxim = 0;
    for(i = 1; i <= n; i++){
        numar = 0;
        for(j = 1; j <= m; j++)
            if(a[i][j] != 0) numar++;
        if(maxim < numar) maxim = numar;
    }
    for(i = 1; i <= n; i++)
    {
        numar=0;
```

```
        for(j = 1; j <= m; j++)
            if(a[i][j] != 0) numar++;
        if(maxim == numar)
            cout<<"\nLinia "<<i<<" are "<<maxim<<"
elemente nenule";
    }
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

Online C++ Compiler IDE

```
1 #include <iostream>
2 using namespace std;
3 int a[30][30], n, m, i, j, maxim, numar;
4 int main(void)
5 {
6     cout<<"Dati dimensiunile matricei A:\n";
7     cin>>n>>m;
8     for(i = 1; i <= n; i++)
9         for(j = 1; j <= m; j++)
10             cin>>a[i][j];
11     cout<<"Elementele matricei A sunt: \n";
12     for(i = 1; i <= n; i++)
13     {
14         for(j = 1; j <= m; j++) cout<<a[i][j]<<" ";
15         cout<<"\n";
16     }
17     maxim = 0;
18     for(i = 1; i <= n; i++)
19     {
20         numar = 0;
21         for(j = 1; j <= m; j++)
22             if(a[i][j] != 0) numar++;
23         if(maxim < numar) maxim = numar;
24     }
25     for(i = 1; i <= n; i++)
26     {
27         numar=0;
28         for(j = 1; j <= m; j++)
29             if(a[i][j] != 0) numar++;
30         if(maxim == numar)
31             cout<<"\nLinia "<<i<<" are "<<maxim<<" elemente nenule";
32     }
33     return 0;
34 }
35
```

Execute Mode, Version, Inputs & Arguments

GCC 9.1.0 ☐ Interactive Stdin Inputs:

CommandLine Arguments:

2 3
1 0 13
0 15 0

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3304 kilobyte(s)

Dati dimensiunile matricei A
Elementele matricei A sunt:
1 0 13
0 15 0

Linia 1 are 2 elemente nenule

Problema 6: Se dă o matrice cu n linii și m coloane și elemente numere naturale. Determinați suma valorilor pare din matrice.

Date de intrare: Programul citește de la tastatură numerele n și m , iar apoi $n*m$ numere naturale, separate prin spații, reprezentând elementele matricei, linie cu linie.

Date de ieșire: Programul afișează pe ecran suma căutată S .

Restricții și precizări: $1 \leq m, n \leq 100$; elementele matricei vor fi mai mici decât 10000

Exemplu

Date de intrare

4 6

4 20 15 23 18 9

1 8 23 22 14 18

17 15 13 18 12 15

3 18 8 20 12 5

Date de ieșire

192

Solutie:

```
#include <iostream>
using namespace std;
int a[101][101], m, n, i, j, s;
int main()
{
    cin >> n >> m;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++)
            cin >> a[i][j];
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++)
            if(a[i][j] % 2 == 0)
                s += a[i][j];
    cout << s;
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 int a[101][101], n, m, i, j, s;
4 int main()
5 {
6     cin >> n >> m;
7     for(i = 1; i <= n; i++)
8         for(j = 1; j <= m; j++)
9             cin >> a[i][j];
10    for(i = 1; i <= n; i++)
11        for(j = 1; j <= m; j++)
12            if(a[i][j] % 2 == 0)
13                s += a[i][j];
14    cout << s;
15    return 0;
16 }
```

Executa Mode, Version, Inputs & Arguments

GCC 9.1.0 ☐ Interactive

Stdin Inputs:

```
4 6
4 20 15 23 18 9
1 8 23 22 14 18
17 15 13 18 12 15
3 18 8 20 12 5
```

CommandLine Arguments:

Result

CPU Time: 0.00 sec(s), Memory: 3276 kilobyte(s)

```
192
```

Problema 7: Se dă o matrice cu n linii și m coloane și elemente numere naturale. Să se determine câte dintre elementele situate pe linii cu indici pari sunt prime.

Date de intrare: Programul citește de la tastatură numerele n și m , iar apoi n șiruri cu câte m numere naturale, reprezentând elementele matricei.

Date de ieșire: Programul va afișa pe ecran numărul C , reprezentând valoarea căutată.

Restricții și precizări: $1 \leq n, m \leq 100$;

elementele matricei sunt numere naturale mai mici decât 1.000.000

liniile matricei sunt numerotate de 1 la n , iar coloanele de la 1 la m

Exemplu

Intrare

4 3

5 12 10

3 9 1

7 10 1

10 9 3

Ieșire

2

Explicație

Cele 2 valori prime determinate sunt cele îngroșate mai jos:

5	12	10
3	9	1
7	10	1
10	9	3

Observăm că matricea conține și alte elemente prime, dar ele nu sunt situate pe linii cu indici pari.

Soluție:

```
#include <iostream>
using namespace std;
long long a[101][101],m,n,i,j,prim,nr,k;
int main()
{
    cin>>n>>m;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= m; j++)
            cin>>a[i][j];
    for(i = 1; i <= n; i++)
        if(i % 2 == 0)
            for(j = 1; j <= m; j++)
            {
                prim = 1;
                if(a[i][j] < 2) prim = 0;
                for(k = 2; k * k <= a[i][j] && prim; k++)
                    if(a[i][j] % k == 0) prim = 0;
                if(prim != 0) nr++;
            }
    cout<<nr;
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 long long a[101][101], m, n, i, j, prim, nr, k;
4 int main()
5 {
6     cin >> n;
7     for(i = 1; i <= n; i++)
8         for(j = 1; j <= m; j++)
9             cin >> a[i][j];
10    for(i = 1; i <= n; i++)
11        if(a[i][1] == a[i][2])
12            for(j = 1; j <= m; j++)
13                if(a[i][j] != a[i][1])
14                    prim = 1;
15    if(prim == 0)
16        for(k = 2; k <= n; k++)
17            if(a[k][1] == a[k][2])
18                if(prim == 0)
19                    nr++;
20    cout << nr;
21    return 0;
22 }
```

Execute Mode, Version, Inputs & Arguments

GCC 8.1.0 Interactive

Settin Inputs:

```
4 3
5 12 10
3 9 1
7 10 1
10 9 3
```

CommandLine Arguments

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3384 kilobyte(s)

```
2
```

Problema 8: Se dă o matrice cu n linii și m coloane și elemente numere naturale. Să se determine câte linii ale matricei au toate elementele egale.

Date de intrare: Programul citește de la tastatură numerele n și m, iar apoi n*m numere naturale, separate prin spații, reprezentând elementele matricei, linie cu linie.

Date de ieșire: Programul afișează pe ecran numărul C, cu semnificația de mai sus.

Restricții și precizări: $1 \leq m, n \leq 100$; elementele matricei vor fi mai mici decât 1.000.000

Exemplu

Date de intrare

4 6

23 23 15 23 18 23

17 17 17 17 17 17

17 18 13 18 12 15

34 34 34 34 34 34

Date de ieșire

2

Explicații

A doua și a patra linie au toate valorile egale.

Solutie:

```
#include <iostream>
using namespace std;
long long a[101][101],n,m, numar_linii_egale, egale;
int main()
{
    cin>>n>>m;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= m; j++)
            cin >> a[i][j];
    for (int i = 1; i <= n; i++)
    {
        egale=1;
        for (int j = 1; j < m; j++)
            for(int k = j + 1; k <= m; k++)
                if(a[i][j]!=a[i][k]) egale=0;
        if(egale) numar_linii_egale++;
    }
    cout<<numar_linii_egale;
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

The screenshot displays the JDoodle online C++ compiler interface. At the top, the C++ code from the previous block is pasted into the editor. Below the editor, the 'Execute Mode, Version, Inputs & Arguments' section is visible. The compiler version is set to 'GCC 9.1.0'. The 'Interactive' checkbox is unchecked. The 'Stdin Inputs' section contains the input values: 4, 5, 23, 23, 15, 23, 18, 23, 17, 17, 17, 17, 17, 17, 17, 18, 13, 18, 12, 15, 34, 34, 34, 34, 34, 34. The 'Execute' button is highlighted in blue. Below the execution controls, the 'Result' section shows the output: '2'. The CPU time is 0.00 seconds and the memory usage is 3284 kilobytes.

```
1 #include <iostream>
2 using namespace std;
3 long long a[101][101],n,m, numar_linii_egale, egale;
4 int main()
5 {
6     cin>>n>>m;
7     for (int i = 1; i <= n; i++)
8         for (int j = 1; j <= m; j++)
9             cin >> a[i][j];
10    for (int i = 1; i <= n; i++)
11    {
12        egale=1;
13        for (int j = 1; j < m; j++)
14            for(int k = j + 1; k <= m; k++)
15                if(a[i][j]!=a[i][k]) egale=0;
16        if(egale) numar_linii_egale++;
17    }
18    cout<<numar_linii_egale;
19    return 0;
20 }
```

Execute Mode, Version, Inputs & Arguments

GCC 9.1.0

Interactive

Stdin Inputs

4 5
23 23 15 23 18 23
17 17 17 17 17 17
17 18 13 18 12 15
34 34 34 34 34 34

Execute

Result

CPU Time: 0.00 secs, Memory: 3284 kilobytes

2

Problema 9: Se dă o matrice cu n linii și n coloane și elemente numere naturale. Calculați cel mai mare divizor comun al sumei elementelor de deasupra diagonalei principale și al sumei elementelor de sub diagonala principală.

Date de intrare: Programul citește de la tastatură numărul n , iar apoi $n*n$ numere naturale, separate prin spații, reprezentând elementele matricei, linie cu linie.

Date de ieșire: Programul afișează pe ecran numărul D , reprezentând valoarea calculată.

Restricții și precizări: $1 \leq n \leq 20$;

elementele matricei vor fi mai mici decât 1.000.000

cel puțin un element situat deasupra diagonalei principale și cel puțin un element situat sub diagonala principală sunt nenule

Exemplu

Date de intrare

4

8 3 5 6

5 5 6 5

3 8 6 5

8 4 8 8

Date de ieșire

6

Explicație

Suma elementelor de sub diagonala principală este 36 iar cea a elementelor de deasupra diagonalei principale este 30. Cel mai mare divizor comun pentru 36 și 30 este 6.

Soluție:

```
#include <iostream>
using namespace std;
long long a[21][21], n, i, j, suma_deasupra_diag_princ,
suma_sub_diag_princ, rest;
int main()
{
    cin >> n;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            cin >> a[i][j];
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            if(i>j) // elementele de deasupra diag. principala
                suma_deasupra_diag_princ += a[i][j];
    else
```



```
        if(i<j)           // elementele de sub diag. secundara
            suma_sub_diag_princ+=a[i][j];
    rest = suma_deasupra_diag_princ % suma_sub_diag_princ;
    while(rest)
    {
        suma_deasupra_diag_princ = suma_sub_diag_princ;
        suma_sub_diag_princ = rest;
        rest = suma_deasupra_diag_princ % suma_sub_diag_princ;
    }
    cout<<suma_sub_diag_princ;
    return 0;
}
```

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

The screenshot displays the JDoodle online C++ compiler interface. At the top, the C++ code is shown in a text editor with line numbers 1 through 26. The code implements a program to calculate the sum of elements on the secondary diagonal of a matrix. Below the code editor, there is a section for 'Execute Mode, Version, Inputs & Arguments'. It includes a dropdown menu for the compiler version (GCC 9.1.0), a checkbox for 'Interactive' mode, and a text area for 'Stdin Inputs' containing the values 4, 8 3 5 6, 5 5 6 5, 3 8 6 5, and 8 4 8 8. There is also a 'CommandLine Arguments' text area. A blue 'Execute' button is visible. Below the execution controls, the 'Result' section shows the CPU Time (0.00 sec(s)) and Memory (3424 kilobyte(s)). The output area at the bottom is a black box with the number 6 displayed.

```
1 #include <iostream>
2 using namespace std;
3 long long a[21][21], n, i, j, suma_deasupra_diag_princ, suma_sub_diag_princ, rest;
4 int main()
5 {
6     cin>>n;
7     for(i=1;i<=n;i++)
8         for(j=1;j<=n;j++)
9             cin>>a[i][j];
10    for(i=1;i<=n;i++)
11        for(j=1;j<=n;j++)
12            if(i>j)           // elementele de deasupra diag. principala
13                suma_deasupra_diag_princ+=a[i][j];
14            else
15                if(i<j)       // elementele de sub diag. secundara
16                    suma_sub_diag_princ+=a[i][j];
17    rest = suma_deasupra_diag_princ % suma_sub_diag_princ;
18    while(rest)
19    {
20        suma_deasupra_diag_princ = suma_sub_diag_princ;
21        suma_sub_diag_princ = rest;
22        rest = suma_deasupra_diag_princ % suma_sub_diag_princ;
23    }
24    cout<<suma_sub_diag_princ;
25    return 0;
26 }
```

Execute Mode, Version, Inputs & Arguments

GCC 9.1.0 Interactive Stdin Inputs

4
8 3 5 6
5 5 6 5
3 8 6 5
8 4 8 8

CommandLine Arguments

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3424 kilobyte(s)

6

Problema 10: Se dă o matrice pătratică cu n linii și n coloane și elemente numere naturale mai mici decât 1000. Să se afișeze în ordine strict crescătoare valorile care apar sub diagonala principală și sub diagonala secundară de cel puțin 2 ori. Fiecare valoare se va afișa o singură dată.

Date de intrare: Programul citește de la tastatură numărul n , apoi elementele matricei, linie cu linie.

Date de ieșire: Programul va afișa pe ecran, în ordine strict crescătoare, valorile cerute, separate printr-un spațiu.

Restricții și precizări: $1 \leq n \leq 200$

Exemplu

Intrare

```
6
10 8 5 8 4 2
6 5 3 1 3 8
8 1 4 7 8 8
5 1 9 6 6 1
8 9 3 2 3 6
8 9 3 3 9 6
```

Ieșire

```
3 9
```

Solutie:

```
#include <iostream>
using namespace std;
int n, i, j, k, nr, a[201][201], aparitii[1001], valoare_maxima=-1001;
int main()
{
    cin>>n;
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
            cin>>a[i][j];
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
        {
            if ( i > j && i+j > n + 1 ) aparitii[a[i][j]]++;
            if(valoare_maxima < a[i][j]) valoare_maxima = a[i][j];
        }
    for(i=1; i<=valoare_maxima; i++)
        if(aparitii[i]>=2) cout<<i<<" ";
    return 0;
}
```

Laborator - Programarea Calculatoarelor si Limbaje de Programare (2022) Limbajul C++

Adrian Runceanu

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 int n, i, j, k, nr, a[201][201], aparitii[1001], valoare_maxima=-1001;
4 int main()
5 {
6     cin>>n;
7     for (i=1; i<=n; i++)
8         for (j=1; j<=n; j++)
9             cin>>a[i][j];
10    for (i=1; i<=n; i++)
11        for (j=1; j<=n; j++)
12        {
13            if ( i > j && i+j > n + 1 ) aparitii[a[i][j]]++;
14            if(valoare_maxima < a[i][j]) valoare_maxima = a[i][j];
15        }
16    for(i=1; i<=valoare_maxima; i++)
17        if(aparitii[i]>=2) cout<<i<<" ";
18    return 0;
19 }
```

Execute Mode, Version, Inputs & Arguments

GCC 9.1.0 ☐ Interactive Stdin Inputs

6
10 8 5 8 4 2
6 5 3 1 3 8
8 1 4 7 8 8
5 1 9 6 6 1
8 9 3 2 3 6
8 9 3 3 9 6

CommandLine Arguments

Result

CPU Time: 0.00 sec(s), Memory: 3396 kilobyte(s)

3 9

B. Probleme propuse spre rezolvare:

L6_1). Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) avand componente numere intregi. Se cere sa se calculeze suma componentelor de pe marginea (rama) matricei.

Exemplu: Pentru $n = 3$ si $m = 4$ si

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 10 & 11 & 12 & 5 \\ 9 & 8 & 7 & 6 \end{pmatrix}$$

se va afisa suma $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 47$.

L6_2). Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) avand componente numere intregi. Sa se determine maximul fiecărei coloane și minimul fiecărei linii.

L6_3). Să se construiască o matrice pătratică ($n = m$) de dimensiune n^2 ($1 \leq n \leq 30$) cu primele numere pare incepand cu 2.

L6_4). Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) cu elemente numere reale. Sa se scrie un program C++ care sa schimbe, in aceasta matrice, prima linie cu ultima linie. Se va afișa apoi matricea modificata.

L6_5). Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) cu elemente numere reale. Sa se scrie un program C++ care sa schimbe, in aceasta matrice, prima coloana cu ultima coloana. Se va afișa apoi matricea modificata.

L6_6). Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) cu elemente numere reale. Sa se scrie un program C++ care sa determine un vector care sa conțină elementele maxime de pe fiecare linie a matricei, iar apoi sa determine cea mai mica valoare din vectorul construit anterior.

L6_7). Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) cu elemente numere reale. Sa se scrie un program C++ care sa determine un vector care sa conțină elementele minime de pe fiecare coloană a matricei, iar apoi sa determine cea mai mare valoare din vectorul construit anterior.

Bibliografie

- [1] <http://www.pbinfo.ro> Descrierea site-ului: "*www.pbinfo.ro* îți propune să rezolvi probleme de informatică, cu evaluator automat. Știi pe loc dacă soluția ta este corectă sau dacă trebuie să mai lucrezi la ea. Problemele sunt grupate după programa de informatică pentru liceu. Dar nu trebuie să fii la liceu ca să rezolvi aceste probleme. Poți fi elev de gimnaziu, student, profesor sau pur și simplu pasionat de informatică. De fapt, trebuie doar să vrei!!"
- [2] <https://www.runceanu.ro/adrian>
- [3] Adrian Runceanu – „*Programarea și utilizarea calculatoarelor*”, Editura Academica Brâncuși din Târgu-Jiu, 2003, ISBN 973-8436-44-3
- [4] Adrian Runceanu, Mihaela Runceanu, „*Noțiuni de programare – limbajul C++*”, Editura Academica Brâncuși din Târgu-Jiu, 2012, ISBN 978-973-144-550-2
- [5] Adrian Runceanu, Mihaela Runceanu, „*Algoritmi implementati in limbajul C++. Volumul I – Algoritmi elementari*”, Editura Academica Brâncuși din Târgu Jiu, 2021, ISBN 978-606-9614-06-8